Version Control: Git and GitHub Essentials

Objective

To introduce beginners to GitHub and demonstrate key operations like configuring Git, cloning repositories, checking status, committing, and pushing changes.

1. What is GitHub? — An Introduction for Beginners

Purpose

GitHub is a **web-based platform** for version control and collaboration using **Git**. It allows developers to host, manage, and share their code efficiently.

Key Features

- **Version Control:** Tracks and manages code changes.
- **Collaboration:** Multiple developers can work on the same project.
- **Hosting:** Provides secure, cloud-based repositories.
- **Project Management:** Includes issues, pull requests, and project boards.

2. How to Register an Account on GitHub

Purpose

To create a GitHub account for managing repositories and collaborating online.

Steps

1. Go to https://github.com/signup
2. Enter your **email, username, and password**.
3. **Verify** your email through the confirmation link sent by GitHub.
4. Personalize your account by answering optional questions.
5. Set up **Two-Factor Authentication (2FA)** for better security.

3. Installing Git on Windows

Purpose

To install Git for version control on your local computer and connect it with GitHub.

Steps

1. Visit the official Git website: https://git-scm.com/downloads
2. Click **Download for Windows** to get the installer file (`Git-x.x.x.exe`).
3. **Run the installer** and follow these steps:
    - Accept the license agreement.
    - Keep the default installation path.
    - Select **"Git Bash Here"** option.
    - Choose a preferred text editor (e.g., Visual Studio Code or Notepad++).
    - Leave the remaining options as default and complete installation.
4. Once installed, open **Git Bash** from the Start menu or by right-clicking on the desktop → *Git Bash Here*.
5. Verify the installation:
6. `git --version`

    Example output:

    ```
    git version 2.45.0.windows.1
    ```

4. Configure Git with Username and Email

Purpose

Before using Git, configure your **username and email**. These appear in your commits and identify your contributions.

Commands

```
git config --global user.name "Your Name"
git config --global user.email "youremail@example.com"
```

Verify your settings:

```
git config --list
```

5. Cloning a Repository

Purpose

To make a local copy of a remote repository for editing or contribution.

Steps

1. Go to the desired repository on GitHub.
2. Click the **Code** button and copy the HTTPS URL.
3. Open **Git Bash** and run:
4. `git clone <repository_url>`
5. Move into the cloned folder:

6. `cd repository_name`

6. Checking Repository Status

Purpose

To view which files are modified, added, or deleted in your working directory.

Command

```
git status
```

Example Output

```
On branch main
Changes not staged for commit:
  modified: index.html
Untracked files:
  newfile.py
```

7. Committing and Pushing Changes

Purpose

To save your work locally and then upload it to GitHub.

Steps

1. Check file status:

   ```
   git status
   ```

2. Add modified files to the staging area:

   ```
   git add .
   ```

3. Commit with a message:

   ```
   git commit -m "Describe your changes"
   ```

4. Push to the GitHub repository:

   ```
   git push
   ```

8. Common Git Commands Summary

Command	Description
`git --version`	Check Git installation
`git config --global user.name "Name"`	Set username
`git config --global user.email "Email"`	Set email address
`git clone <URL>`	Clone a repository
`git status`	Show current changes
`git add .`	Add all changes to staging
`git commit -m "message"`	Save changes locally
`git push`	Push commits to GitHub

Conclusion

You've now learned how to:

- Install Git on Windows
- Create a GitHub account
- Configure Git with your details
- Clone, check, commit, and push repositories

With these foundational steps, you're ready to start collaborating effectively using **Git and GitHub**.